Limbaje de programare

Un limbaj de programare este un set bine definit de expresii și reguli (sau tehnici) valide de formulare a instrucțiunilor pentru un computer. Un limbaj de programare are definite un set de reguli sintactice și semantice. El dă posibilitatea programatorului să specifice în mod exact și amănunțit acțiunile pe care trebuie să le execute calculatorul, în ce ordine și cu ce date. Specificarea constă practic în întocmirea/scrierea programelor necesare ("programare").

Categorii

După paradigma programării

Limbaje ezoterice, ex. Brainfuck;

Limbaje procedurale; ex. C, Java, Perl;

Limbaje functionale: ex. ML, Haskell;

Limbaje pentru programarea logică ex.Prolog, DATALOG;

Limbaje mixte: ex.LISP oferă un amestec heteroclit și netipizat de programare imperativă și programare functională.

După nivelul de abstractizare (care printre altele corespunde cu ușurința cu care programele pot fi citite și înțelese de către programatori):

Limbaje de generația întâia: limbajele cod-mașină (limbaje mașină);

Limbaje de generația a doua: limbajele de asamblare;

Limbaje de generația a treia: limbaje de nivel înalt, cum ar fi C-ul (și toate derivatele sale: C++, Java, ...), Pascal, Fortran ș.a.;

Limbaje de generația a patra: limbajele neprocedurale, orientate pe rezolvarea unei anumite clase de probleme: SQL ș.a.;

Limbaje de generația a cincea: limbaje utilizate în domenii precum logica fuzzy, inteligența artificială sau și rețelele neuronale: Prolog, LISP ș.a.

După vechime

Limbaje din secolul trecut; ex. LISP și FORTRAN din anii '50;

Limbaje din ultimul deceniu; ex. Haskell, Ruby, Brainfuck.

Execuție

Pentru executarea unui program scris într-un limbaj oarecare, există, în principiu, două abordări: compilare sau interpretare. La compilare, compilatorul transformă programul-sursă în totalitatea sa într-un program echivalent scris în limbaj mașină, care apoi este executat. La interpretare interpretorul ia prima instrucțiune din programul-sursă, o transformă în limbaj mașină și o execută; apoi trece la instrucțiunea doua și repetă aceleași acțiuni ș.a.m.d.

Unele limbaje se pretează bine la compilare; de exemplu limbajele clasice: Pascal, Fortran, C. Altele sunt cu predilecție interpretate, de ex. BASIC, PHP, SQL. Multe limbaje moderne combină compilarea cu interpretarea: codul sursă este compilat într-un limbaj binar numit bytecode, care la rulare este interpretat de către o mașină virtuală. De remarcat faptul că unele interpretoare de limbaje pot folosi compilatoare așa-numite just-in-time, care transformă codul în limbaj mașină chiar înaintea executării.

Listă de limbaje de programare

Limbaje generale

Limbaje care pot fi folosite (cel puțin teoretic) la rezolvarea oricărui tip de probleme:

TEO (Basic)

Alen (variantă modificată a limbajului TEO)

BASIC (QuickBasic, GW-BASIC, Visual Basic)

C

C++

C#

Clips

Cobol

Fortran

Haskell

Java

JavaScript

LISP

Pascal (și ObjectPascal, folosit de Borland Delphi)

Perl

PHP

Prolog

Python

Ruby

Seed7

Scriptol

Tcl/TK

XHTML

MATLAB

SmallTalk

D

Borland Delphi

Algol

Simula

Limbaje specifice

Limbaje cu un scop bine definit – de obicei interacțiunea cu un anumit program sau sistem.

ActionScript – limbajul de scripting folosit de Macromedia Flash

ASP (și ASP.NET) – pentru scripting web

FoxPro (FoxPro, Visual FoxPro)

MIVA SCRIPT

SQL - pentru interacțiunea cu bazele de date relaționale

Popularitate limbaje de programare